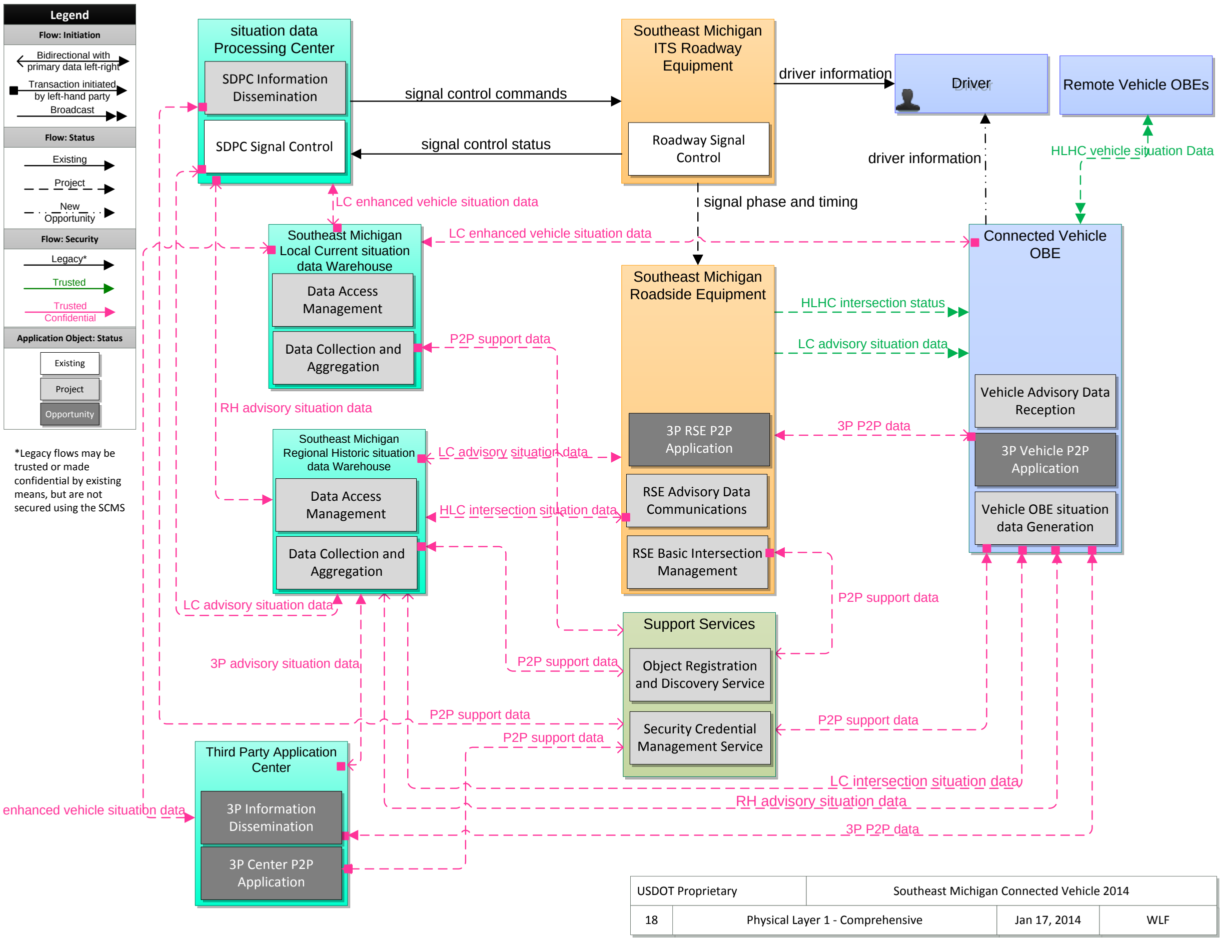
1 General

One of the goals of the Southeast Michigan 2014 project is to investigate the use of a common process for assuring trust in and protecting confidentiality (when appropriate) and integrity of data in transmission used by all applications independent of communication medium or purpose.


Legend
Flow: Initiation
Bidirectional with primary data left-right
Transaction initiated by left-hand party
Broadcast
Flow: Status
Existing
Project
New Opportunity
Flow: Security
Legacy*
Trusted
Trusted Confidential
Application Object: Status
Existing
Project
Opportunity
*Legacy flows may be trusted or made confidential by existing means, but are not secured using the SCMS
situation data Processing Center
SDPC Information Dissemination
SDPC Signal Control
signal control commands
signal control status
Southeast Michigan ITS Roadway Equipment
Roadway Signal Control
driver information
Driver
Remote Vehicle OBEs
driver information
HLHC vehicle situation Data
LC enhanced vehicle situation data
signal phase and timing
Southeast Michigan Local Current situation data Warehouse
Data Access Management
Data Collection and Aggregation
LC enhanced vehicle situation data
Connected Vehicle OBE
Southeast Michigan Roadside Equipment
HLHC intersection status
LC advisory situation data
P2P support data
Vehicle Advisory Data Reception
RH advisory situation data
3P RSE P2P Application
3P P2P data
3P Vehicle P2P Application
Southeast Michigan Regional Historic situation data Warehouse
LC advisory situation data
Data Access Management
RSE Advisory Data Communications
Vehicle OBE situation data Generation
HLC intersection situation data
Data Collection and Aggregation
RSE Basic Intersection Management
LC advisory situation data
P2P support data
Support Services
3P advisory situation data
P2P support data
Object Registration and Discovery Service
P2P support data
Security Credential Management Service
P2P support data
P2P support data
Third Party Application Center
LC intersection situation data
RH advisory situation data
LC enhanced vehicle situation data
3P Information Dissemination
3P P2P data
3P Center P2P Application
USDOT Proprietary
Southeast Michigan Connected Vehicle 2014
18
Physical Layer 1 - Comprehensive
Jan 17, 2014
WLF

1.1 Communication types

We identify two types of communication patterns that support applications: *broadcast* and *transactional*.

Broadcast-supported applications send unencrypted, broadcast messages, which are intended to be consumed by any receiver in the vicinity. Examples of broadcast communications are BSM, SPaT, MAP.[1]

[1] Note that this definition identifies an application with a message set, rather than with a particular process on the computing platform or a particular set of user interactions.

NOTE: In the Southeast Michigan Test Bed, all broadcast messages will be signed immediately before the final transmission. In other words, if a message originates at a server and is sent to an RSE for broadcast, it will be signed by the RSE, not the server.

Transactional-type applications are exchanges between two objects for the purposes of carrying out some transaction. In the Southeast Michigan Test Bed, the concept of operations is that transactional applications will consist of request-response activities with small data transfers (up to approx. 10 Kbytes). The Southeast Michigan Test Bed concept of operations does not cover larger data transfers that need to be handed off between multiple RSU sessions.

This set of security requirements does not consider groupcast communications, that is, applications that involve communications between groups of more than two devices but are not broadcast to everyone. For example, applications that use a publish-subscribe mechanism might naturally use groupcast. The security framework for the Southeast Michigan Test Bed will not natively support groupcast.

This set of security requirements does not consider the need for device physical security.

This set of security requirements does not address data protection at endpoints, for example encryption of databases. It is, however, assumed that endpoints that store Personal Identifiable Information (PII) shall take appropriate measures to protect that PII.

1.2 Privacy considerations for a multi-application setting

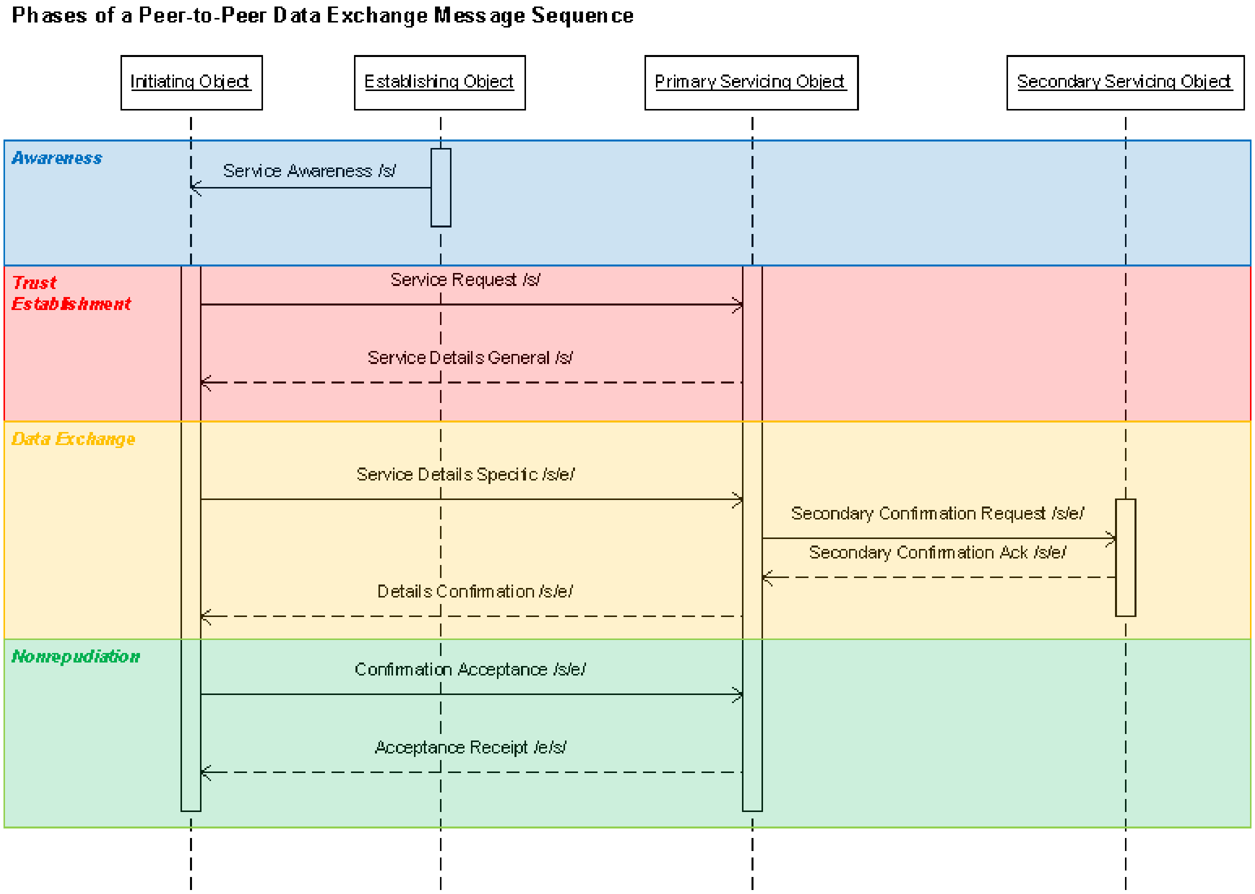
The system requirements are formulated so as to preserve the anonymity of travelers to the greatest extent practicable. In a multi-application setting this is particularly challenging, because (a) the traveler may have greater privacy protection/preservation expectations than those integrated into specific application, and (b) correlations between applications may reveal information about a traveler not present in a single application – in other words, from an attacker learning information about the traveler not just from the contents of individual application messages, but from the fact that one user is running two applications.

Some applications by their nature will require traveler or vehicle-specific information: for example, BSMs reveal vehicle location. This makes it all the more important to ensure that applications do not use or exchange this information unless it is absolutely necessary, as revealing the information within application A may allow it to be correlated with information from application B.

The requirements below are written bearing in mind the need to limit the collection and use of information about or relating to a specific vehicle or traveler against correlation of data and metadata from multiple applications.

Further discussion of privacy and security for the multi-application setting can be found in EU-US ITS Task Force Standards Harmonization Working Group Harmonization Task Group 1 report 1-1, "Current Status of Security Standards", section 14 and Annex C.

2 Requirements for transactional unicast communications


Phases of a Peer-to-Peer Data Exchange Message Sequence
Initiating Object
Establishing Object
Primary Servicing Object
Secondary Servicing Object
Awareness
Service Awareness /s/
Trust Establishment
Service Request /s/
Service Details General /s/
Data Exchange
Service Details Specific /s/e/
Secondary Confirmation Request /s/e/
Secondary Confirmation Ack /s/e/
Details Confirmation /s/e/
Nonrepudiation
Confirmation Acceptance /s/e/
Acceptance Receipt /e/s/

Participants are identified as User and Server. The User initiates the exchange. The User wants to engage in a service offered by the Server – this service could be that the Server provides data to the User, that the User provides data to the Server to store, or that the Server and User exchange data. There may be multiple distinct instances of a Server for a given system, and one instance may be better than the others – we call this the preferred instance.

1) Service discovery:
   a. The exchange shall allow the User to determine whether it is communicating with the preferred instance of the given server.
   b. No server-side service discovery requirements.
2) Authorization
   a. The exchange shall allow the User to demonstrate that they are authorized to use the service.
   b. The exchange shall allow the Server to demonstrate that they are authorized to provide the service.
   c. The definition of “authorized to use the service” will be application specific. Examples of what might be demonstrated to show authorization include:
      (a) that the object is of the right physical type

(b) that it has the desired software installed

(c) that it does not have bad software installed

(d) that it currently has appropriate credentials, including the correct PSID (and possibly SSP) if those credentials are a 1609.2 certificate

(e) that it is in an approved physical location

3) Privacy

a. The exchange shall not require either party to exchange unencrypted data. Sensitive information is any information beyond what is necessary to establish authorization as described above.

b. Information exchanges should be limited to that necessary to establish service authorizations and permissions.

(a) User location information will only be exchanged when as part of service provision or necessary for the server to verify that the user is authorized to use the service, for example to prevent a service from being participated in by the wrong User.

c. The exchange shall not use identifiers that can be directly linked to the User's physical identity (VIN, license number, etc.).

d. The exchange shall, as far as practical, use temporary and one-time identifiers. Separate instances of the exchange shall, as far as practical, not use identifiers (User MAC address, UE-ID (IMEI)[2], IP address, certificate, temporary ID, session ID, etc.) that have been used in a previous instance of the exchange.

4) Integrity.

a. Each party to the exchange shall be satisfied that messages, including any metadata necessary to carry out the exchange correctly, have not been modified since they were created by the other party.[3]

5) Replay / message order

a. The exchange shall guarantee that messages from both parties are fresh and have not been replayed.

b. For every message in the exchange that is a response to a particular previous message, the exchange shall ensure that the response message is recognized as a response to the correct previous message and cannot be incorrectly associated with a different previous message.

6) Non-repudiation / Audit

a. If one object provided incomplete or incorrect information, it shall be possible to prove this to a third party using a transcript of the exchange (in other words, it shall not be possible for one object to forge evidence that the other object provided incorrect information).

[2] We will investigate whether 3GPP standards support changing the network identifier on request from applications as is done with the MAC address of 802.11 defined devices.

[3] Here, the exchange being "carried out correctly" is defined as either the correct outcome happens, or the packet gets lost.

7) Performance
   a. The security processing shall add as few round trips as possible to the exchange.
   b. The security processing shall not require either object to communicate in real time with a third party, for example to get keys or revocation information.
8) Removal of misbehaving objects
   a. For each given application setting, there shall be an object or subsystem that is able to correctly determine that an object in that application setting is misbehaving, i.e. if that is sending information that is wrong in a significant or persistent way (where “significance” may be application-specific). This shall include protection against false accusations of misbehavior.
   b. If an object is misbehaving, it shall be possible to prevent the object’s information from being trusted in the future, either by denying it new credentials or by revoking its existing credentials.
   c. For settings in which revocation is used to remove misbehaving objects, it shall be possible for a User to efficiently check the revocation status of a Server, and vice-versa.
   d. If an object is itself revoked, it shall be possible for the object to discover that it is revoked. Mechanisms to recover from revocation are out of scope for the Southeast Michigan Test

3 Requirements for broadcast applications

The application that sends a broadcast message is referred to as the User. The application that receives a broadcast message is referred to as the Receiver.

1) Service discovery: No requirements.
2) Authorization: The security process shall allow the User to demonstrate that they are authorized to send the message.
   a. The definition of “authorized to send the message” will be application specific. Examples of what might be demonstrated to show authorization include:
      (a) that the object is of the right physical type
      (b) that it has the desired software installed
      (c) that it does not have bad software installed
      (d) that it currently has appropriate credentials, including the correct PSID (and possibly SSP) if those credentials are a 1609.2 certificate
      (e) that it is in an approved physical location
3) Privacy
   a. The exchange shall not require the User to reveal sensitive information unencrypted. Sensitive information is any information beyond what is necessary to establish authentication as described in the message sequence above. Where possible the information exchange shall limit data exchange to the revelation of User’s permissions

information rather than about its identity, unless its identity is explicitly required to determining whether or not it was allowed to send the message.

b. The message shall not contain the User's location information unless this is a necessary part of the application data, or necessary to demonstrate that the User is entitled to send the message.

c. The message shall not use identifiers that linked or linkableto the User's real-world identity (VIN, license number, etc.).

d. The message shall, as far as practical, use temporary and one-time identifiers. Separate instances of the message shall, as far as practical, not use identifiers (User MAC address, IP address, certificate, temporary ID, session ID, etc.) that have been used in a previous instance of the exchange.

4) Integrity. The Receiver shall be satisfied that messages have not been modified since they were created by the User.

5) Replay / message order: The security process shall guarantee that the User's messages are fresh and have not been replayed.

6) Non-repudiation / Audit: If the User provides incorrect information, it shall be possible to prove this to a third party using a transcript of the exchange (in other words, it shall not be possible for one party to forge evidence that the other party provided incorrect information).

7) Performance: The security process shall not require the User or Receiver to communicate in real time with a third party, for example to get keys or revocation information.

8) Removal of misbehaving objects

a. For each given application setting, there shall be an object or subsystem that is able to determine that an object in that application setting is misbehaving, i.e. if that is sending information that is wrong in a significant or persistent way (where "significance" may be application-specific)

b. If an object is misbehaving, it shall be possible to prevent the object's information from being trusted in the future, either by denying it new credentials or by revoking its existing credentials.

c. For settings in which revocation is used to remove misbehaving objects, it shall be possible for a User to efficiently check the revocation status of a Server, and vice-versa.

d. If an object is itself revoked, it shall be possible for the object to discover that it is revoked. Mechanisms to recover from revocation are out of scope for the Southeast Michigan Test